# Text Summarization

## Find Text

import pysrt  
import pandas as pd  
import re  
from sentence\_transformers import SentenceTransformer  
# install faiss-cpu  
import faiss  
import time  
subs = pysrt.open("The.Batman.2022.1080p.WEB-DL.DDP5.1.Atmos.H.264-EVO-HI.srt")  
  
DF = pd.DataFrame([  
 {  
 "Text": sub.text  
} for sub in subs])  
  
DF  
def remove\_noise(text):  
 text = re.sub("<.\*>", " ", text)  
 text = re.sub("{.\*}", " ", text)  
 text = re.sub("\[.\*\]", " ", text)  
 text = text.strip()  
 return text  
  
DF['clean'] = DF['Text'].apply(remove\_noise)  
  
DF = DF[DF['clean'] != ""]  
  
DF

Text \  
0 <font color="#ffff00">Provided by explosivesku...   
11 [man] Hey!   
14 <i>Good evening and welcome to\nGC-1 News live...   
16 <i>just-released polls have\nincumbent Mayor D...   
17 <i>and 28-year-old grassroots challenger\nBell...   
... ...   
2404 You should go.   
2405 Selina...   
2406 Take care of yourself.   
2414 <font color="#ffff00">Provided by explosivesku...   
2418 - [beep]\n- [static crackles]   
  
 clean   
0 https://twitter.com/kaboomskull   
11 Hey!   
14 Good evening and welcome to\nGC-1 News live at...   
16 just-released polls have\nincumbent Mayor Don ...   
17 and 28-year-old grassroots challenger\nBella R...   
... ...   
2404 You should go.   
2405 Selina...   
2406 Take care of yourself.   
2414 https://twitter.com/kaboomskull   
2418 - \n-   
  
[1573 rows x 2 columns]

Create A Search Engine Using each sentence as your “documents”, create a search engine to find specific pieces of text.

# this is creating the embeddings   
model = SentenceTransformer('msmarco-MiniLM-L-12-v3')  
bodies\_text\_embds = model.encode(DF['clean'].to\_list())  
# Create an index using FAISS  
index = faiss.IndexFlatL2(bodies\_text\_embds.shape[1])  
index.add(bodies\_text\_embds)  
faiss.write\_index(index, 'index\_bodies')  
  
bodies\_text\_embds  
# define a search   
def search(query, k):  
   
 t=time.time()  
 query\_vector = model.encode([query])  
 top\_k = index.search(query\_vector, k)  
 print('totaltime: {}'.format(time.time()-t))  
 return [DF['clean'].to\_list()[\_id] for \_id in top\_k[1].tolist()[0]]

C:\Users\user7\AppData\Roaming\Python\Python39\site-packages\huggingface\_hub\file\_download.py:149: UserWarning: `huggingface\_hub` cache-system uses symlinks by default to efficiently store duplicated files but your machine does not support them in C:\Users\user7\.cache\huggingface\hub\models--sentence-transformers--msmarco-MiniLM-L-12-v3. Caching files will still work but in a degraded version that might require more space on your disk. This warning can be disabled by setting the `HF\_HUB\_DISABLE\_SYMLINKS\_WARNING` environment variable. For more details, see https://huggingface.co/docs/huggingface\_hub/how-to-cache#limitations.  
To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to see activate developer mode, see this article: https://docs.microsoft.com/en-us/windows/apps/get-started/enable-your-device-for-development  
 warnings.warn(message)

Search for several items.

search("cop", 10)  
search("gun", 10)  
search("car", 10)

totaltime: 0.0249330997467041  
totaltime: 0.019948482513427734  
totaltime: 0.02592778205871582

['Her body was in his car.',  
 "Yeah, he's got a car.",  
 'Get out of the car!',  
 'What?',  
 'What?',  
 'What?',  
 'What?',  
 'What?',  
 'Get out of the car and show your hands!',  
 '- \n-']